

# QSM368ZP-WF&SG368Z 系列

## Linux&Ubuntu&OpenWrt

### 显示驱动开发指导

智能产品

版本：1.2

日期：2025-12-12

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<https://www.quectel.com.cn/contact>。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：

<https://www.quectel.com.cn/contact?tab=t> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 © 上海移远通信技术股份有限公司 2025，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2025.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2023-11-17	Baron Zheng	文档创建
1.0	2024-07-22	Colin Zhang/ Baron Zheng	受控版本
1.1	2025-02-27	Randy Xi	新增 OpenWrt 操作系统及相关的路径、文件等内容。
1.2	2025-12-12	Jay Cheng	新增 Kernel 6.1 相关路径、文件等。

## 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	5
图片索引 .....	6
<b>1 引言 .....</b>	<b>7</b>
<b>2 显示框架简介.....</b>	<b>8</b>
<b>3 显示接口配置.....</b>	<b>10</b>
3.1. MIPI .....	10
3.1.1. MIPI 节点配置.....	10
3.1.2. VOP 的挂载 .....	11
3.1.3. dsi-panel 配置.....	11
3.1.4. 屏幕命令格式说明 .....	14
3.2. eDP .....	14
3.2.1. eDP 节点配置 .....	14
3.2.2. VOP 的挂载 .....	16
3.2.3. edp-panel 配置 .....	16
3.3. HDMI .....	18
3.3.1. HDMI 节点配置.....	18
3.3.2. VOP 的挂载 .....	19
3.4. LVDS .....	19
3.4.1. LVDS 节点配置.....	19
3.4.2. VOP 的挂载 .....	20
3.4.3. lvds-panel 配置.....	20
3.4.4. 数据映射 .....	21
<b>4 显示功能配置.....</b>	<b>24</b>
4.1. 背光配置 .....	24
4.1.1. 背光节点属性介绍 .....	24
4.1.2. 背光配置流程.....	26
4.2. Logo 配置 .....	27
4.2.1. Logo 节点介绍 .....	27
4.2.2. Logo 节点配置 .....	28
4.2.3. Logo 图片加载 .....	29
4.2.4. Logo 要求 .....	29
<b>5 附录 术语缩写.....</b>	<b>30</b>

## 表格索引

表 1: panel 属性介绍 .....	13
表 2: eDP 热插拔属性配置 .....	15
表 3: 背光节点属性介绍 .....	25
表 4: Logo 属性介绍 .....	28
表 5: 术语缩写 .....	30

## 图片索引

图 1: 显示子系统架构 .....	8
图 2: 各 VP 支持的最大分辨率 .....	9
图 3: dsi1-panel 节点配置 .....	12
图 4: dsi1-panel 时序配置 .....	12
图 5: 屏幕规格书数据格式 .....	22
图 6: LVDS 数据映射 .....	22
图 7: eDP 背光节点 1 .....	24
图 8: eDP 背光节点 2 .....	25
图 9: eDP 背光原理图 .....	26
图 10: GPIO 配置表 .....	26

# 1 引言

本文档主要介绍如何在移远通信 QSM368ZP-WF 设备和 SG368Z 系列模块上配置 MIPI、eDP、HDMI 和 LVDS 显示接口，以及显示子系统、VOP、panel、背光、Logo 等相关显示内容。

## 备注

1. 本文档适用于运行 Linux 操作系统的 QSM368ZP-WF 设备和运行 Linux 或 Ubuntu 操作系统的 SG368Z 系列模块。
  2. SG368Z 系列模块本身无法预装 OpenWrt 系统，仅提供 SDK 及开发指导供客户二次开发。若有问题，请联系移远通信技术支持。
-



## 2 显示框架简介

显示子系统是负责管理显示输出相关的软硬件的集成系统，包括 VOP、LVDS、MIPI、eDP 和 HDMI 等显示信号输出组件以及与之对应的软件驱动。整体的显示框架如下图所示。一个 VOP 下有多个 VP，每个 VP 下只能挂载一个显示接口。

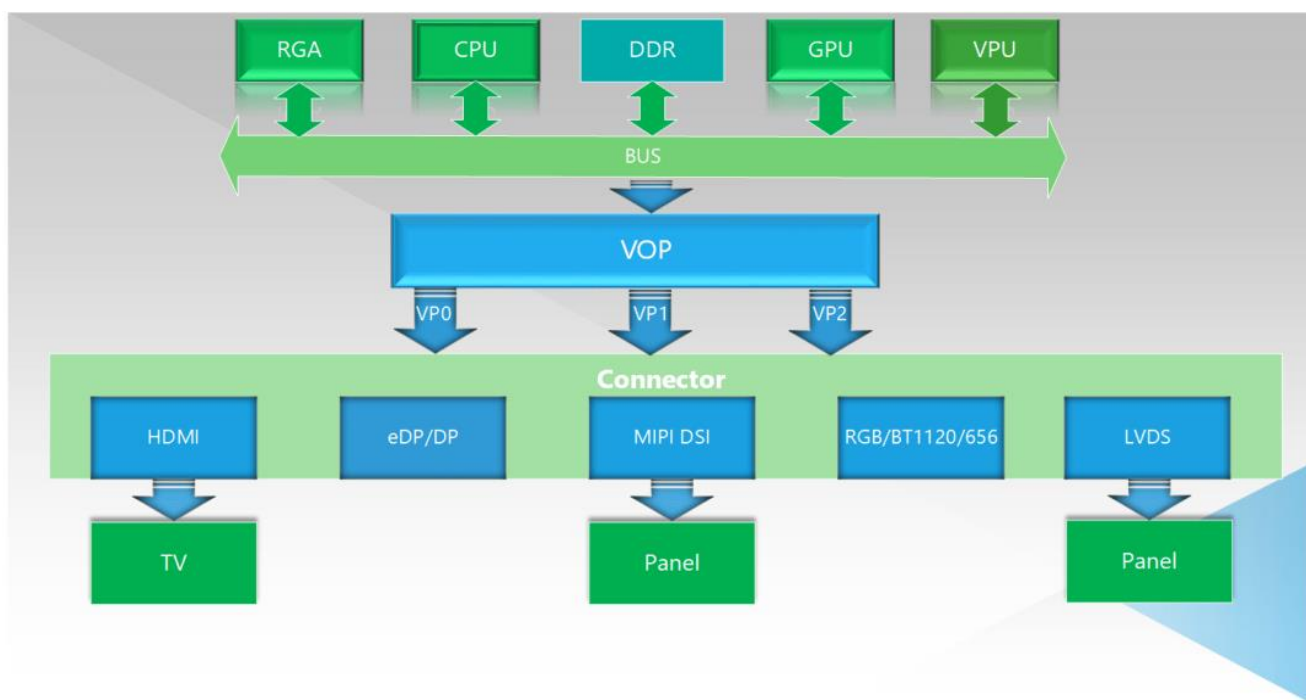


图 1：显示子系统架构

QSM368ZP-WF 设备和 SG368Z 系列模块的 VOP 支持三个 VP（VP0、VP1 和 VP2），最多可支持三屏同显或三屏异显。在配置显示路由时，选择哪个 VOP 作为输入源主要取决于该 VOP 支持的最大分辨率，如下图所示：

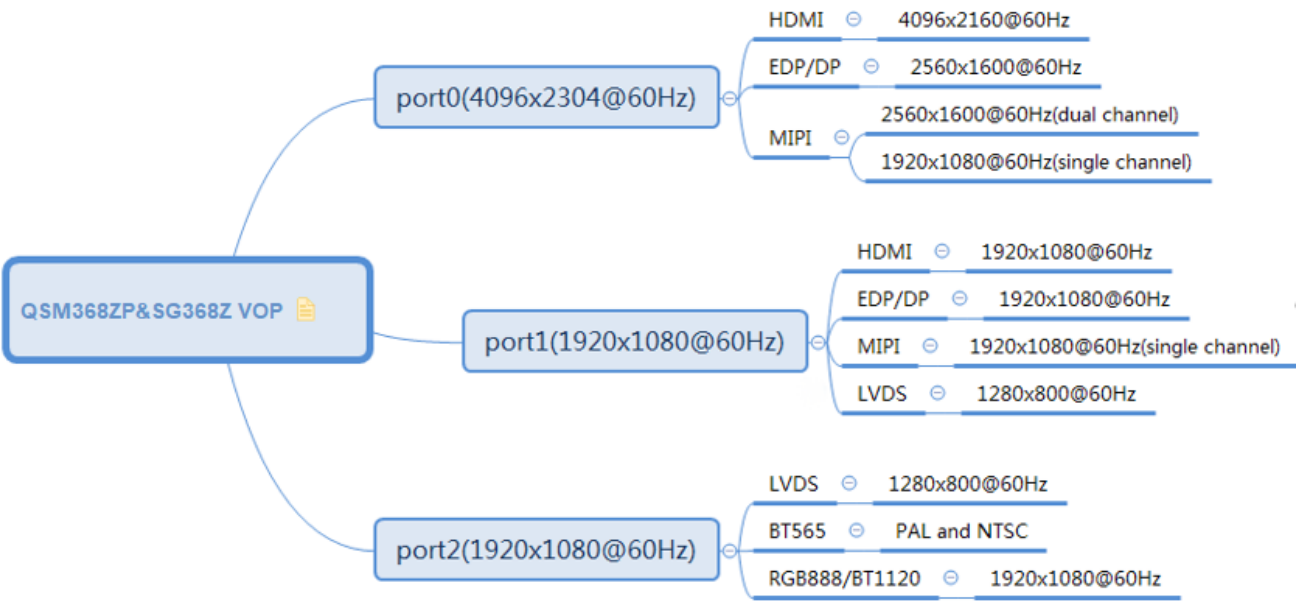


图 2：各 VP 支持的最大分辨率

## 3 显示接口配置

### 3.1. MIPI

#### 3.1.1. MIPI 节点配置

在如下文件中配置 `status = "okay"`将`&dsi1` 节点打开。

- Linux 操作系统:

Kernel 5.10 及之前: `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1: `kernel-6.1/arch/arm64/boot/dts/rockchip/display/dsi1-panel0-ili0881d-720p.dtsi`

- Ubuntu 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi`

```
&dsi1 {
    status = "okay";
};
```

`&dsi1` 依赖`&video_phy1`, 因此还需配置 `status = "okay"`打开`&video_phy1`。

```
&video_phy1 {
    status = "okay";
};
```

### 3.1.2. VOP 的挂载

选择要挂载的 VP，下图以 VP1 为例。当挂载到此 VP 下时，需要关闭其他的 VP 节点，因为一个 VP 下只能挂载一个显示接口，因此需要配置 `status = "disabled"` 将 `&dsi1_in_vp0` 节点关闭，然后配置 `status = "okay"` 将 `&dsi1_in_vp1` 节点打开，反之同理。

```
&dsi1_in_vp0 {
    status = "disabled";
};

&dsi1_in_vp1 {
    status = "okay";
};
```

### 3.1.3. dsi-panel 配置

在如下文件中配置 panel 节点属性。

- Linux 操作系统：

Kernel 5.10 及之前： `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1： `kernel-6.1/arch/arm64/boot/dts/rockchip/display/dsi1-panel0-ili0881d-720p.dtsi`

- Ubuntu 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi`

```

&dsil {
    status = "okay";
    //rockchip, lane-rate = <1000>;
    dsil_panel: panel@0 {
        status = "okay";
        compatible = "simple-panel-dsi";
        reg = <0>;
        num = <0>;
        id = [83 94];
        id-reg = <0x04>;
        backlight = <&backlight1>;
        pinctrl-names = "default";
        pinctrl-0 = <&lcd_rst>;
        power-supply = <&vccalv8_image>;
        reset-gpios = <&gpio3 RK_PC7 GPIO_ACTIVE_LOW>;
        enable-gpios = <&gpio2 RK_PD1 GPIO_ACTIVE_HIGH>;
        init-delay-ms = <20>;
        reset-delay-ms = <60>;
        enable-delay-ms = <60>;
        prepare-delay-ms = <60>;
        unprepare-delay-ms = <60>;
        disable-delay-ms = <60>;
        dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST | MIPI_DSI_MODE_LPM)>;
        dsi,format = <MIPI_DSI_FMT_RGB888>;
        dsi,lanes = <4>;

        panel-init-sequence = [
            .....
        ];

        panel-exit-sequence = [
            ...
        ];
    };
};
    
```

图 3: dsi1-panel 节点配置

```

display-timings {
    native-mode = <&dsil_timing2>;
    dsil_timing2: timing2 {
        clock-frequency = <71000000>;
        hactive = <720>;
        vactive = <1280>;
        hfront-porch = <60>;
        hsync-len = <90>;
        hback-porch = <20>;
        vfront-porch = <24>;
        vsync-len = <4>;
        vback-porch = <24>;
        hsync-active = <0>;
        vsync-active = <0>;
        de-active = <0>;
        pixelclk-active = <1>;
    };
};
    
```

图 4: dsi1-panel 时序配置

表 1: panel 属性介绍

属性	描述
<i>num</i>	panel 的唯一标识，必须是大于等于 0 的整数
<i>id</i>	ID 寄存器的返回值数组，十六进制
<i>id-reg</i>	ID 寄存器地址，1 个字节，需要在前面加 “0x” 表示十六进制
<i>panel-read-id-sequence</i>	如果在读取 <i>id-reg</i> 之前需要发送命令的，将命令填写到 <i>panel-read-id-sequence</i> 数组中，否则就不用配置
<i>backlight</i>	背光控制
<i>power-supply</i>	LCD 供电控制
<i>reset-gpios</i>	Reset 控制脚
<i>enable-gpios</i>	Enable 控制脚
<i>init-delay-ms</i>	<i>init</i> 延时控制，根据屏幕规格书中的时序配置
<i>reset-delay-ms</i>	<i>reset</i> 延时控制，根据屏幕规格书中的时序配置
<i>enable-delay-ms</i>	<i>enable</i> 延时控制，根据屏幕规格书中的时序配置
<i>prepare-delay-ms</i>	<i>prepare</i> 延时控制，根据屏幕规格书中的时序配置
<i>unprepare-delay-ms</i>	<i>unprepare</i> 延时控制，根据屏幕规格书中的时序配置
<i>disable-delay-ms</i>	<i>disable</i> 延时控制，根据屏幕规格书中的时序配置
<i>dsi,flags</i>	MIPI_DSI_MODE_VIDEO (MIPI DSI 工作模式): <ul style="list-style-type: none"> <li>● <i>MIPI_DSI_MODE_VIDEO_BURST</i> 表示视频突发模式</li> <li>● <i>MIPI_DSI_MODE_EOT_PACKET</i> 表示默认在 LP 模式下发送初始化序列</li> <li>● <i>MIPI_DSI_MODE_LPM</i> 表示关闭 EOTP 特性</li> </ul>
<i>dsi,format</i>	像素格式
<i>dsi,lanes</i>	数据通道号。范围：1~4
<i>panel-init-sequence</i>	上电初始化序列
<i>panel-exit-sequence</i>	下电初始化序列

### 3.1.4. 屏幕命令格式说明

命令前 3 个字节（十六进制），分别代表 Data Type、Delay 和 Payload Length。从第四个字节开始的数据代表长度为 Payload Length 的 Payload。

例如：

1. 39 00 04 B9 FF 83 94

Data Type: 0x39 (DCS 长写入)

Delay: 0x00 (0 毫秒)

Payload Length: 0x04 (4 字节)

Payload: 0xb9 0xff 0x83 0x94

2. 05 14 01 29

Data Type: 0x05 (DCS 短写入, 无参数)

Delay: 0x14 (20 毫秒)

Payload Length: 0x01 (1 字节)

Payload: 0x29

### 备注

1. QSM368ZP-WF 设备和 SG368Z 系列模块支持两路 DSI，DSI1 用作 MIPI，DSI0 可以复用为 LVDS 或 MIPI。当前默认 DSI0 用作 LVDS，DSI1 用作 MIPI。
2. panel 中延时控制可参考 `kernel/drivers/gpu/drm/panel/panel-simple.c` 文件中的函数进行配置。
  - *prepare* 时序: `panel_simple_prepare`
  - *unprepare* 时序: `panel_simple_unprepare`
  - *enable* 时序: `panel_simple_enable`
  - *disable* 时序: `panel_simple_disable`

## 3.2. eDP

### 3.2.1. eDP 节点配置

在如下文件中配置 `status = "okay"` 将 `&edp` 节点打开，其中 `force-hpd` 为热插拔使能配置，详见表 2。

- Linux 操作系统:

Kernel 5.10 及之前: `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1: `kernel-6.1/arch/arm64/boot/dts/rockchip/display/lcd-edp-vp0-g160bm-2560x1600.dtsi`

- Ubuntu 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi`

```
&edp {
    force-hpd;
    //hpd-gpios = <&gpio0 RK_PC2 GPIO_ACTIVE_HIGH>;
    status = "okay";
};
```

配置 `status = "okay"`, 打开 `&edp_phy` 节点:

```
&edp_phy {
    status = "okay";
};
```

表 2: eDP 热插拔属性配置

属性	描述
<code>force-hpd</code>	对于嵌入式连接 (Embedded Connection), 一般不需要 HPD 功能, 需要加上该属性。当使能改属性后, <code>hpd-gpios</code> 无效
<code>hpd-gpios</code>	对于盒式连接 (Box-to-box Connection), 一般需要 HPD 功能, 需要配置该属性



### 3.2.2. VOP 的挂载

选择要挂载的 VP，下图以 VP0 为例。当挂载到此 VP 下时，需要关闭其他的 VP 节点，因为一个 VP 下只能挂载一个显示接口，因此需要配置 `status = "disabled"` 将 `&edp_in_vp1` 节点关闭，然后配置 `status = "okay"` 将 `&edp_in_vp0` 节点打开，反之同理。

```
&edp_in_vp0 {
    status = "okay";
};

&edp_in_vp1 {
    status = "disabled";
};
```

### 3.2.3. edp-panel 配置

在如下文件中找到 `edp-panel` 节点进行 panel 属性配置。

- Linux 操作系统：

Kernel 5.10 及之前： `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1： `kernel-6.1/arch/arm64/boot/dts/rockchip/display/lcd-edp-vp0-g160bm-2560x1600.dtsi`

- Ubuntu 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi`

```
edp-panel {
    status = "okay";
    compatible = "simple-panel";
    backlight = <&backlight2>;
    bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
    enable-gpios = <&gpio3 RK_PD1 GPIO_ACTIVE_HIGH>;
    bpc = <8>;
    enable-delay-ms = <20>;
    bl-delay-ms = <20>;
    prepare-delay-ms = <120>;
    unprepare-delay-ms = <120>;
    disable-delay-ms = <20>;
    display-timings {
        native-mode = <&edp_timing>;
        edp_timing: edp_timing {
            clock-frequency = <270000000>;
            hactive = <2560>;
            vactive = <1600>;

            hfront-porch = <48>;
            hsync-len = <32>;
            hback-porch = <112>;

            vfront-porch = <3>;
            vsync-len = <6>;
            vback-porch = <33>;

            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
};
```

panel 的通用属性参考第 3.1.3 章的表 1 进行配置。一般情况下，配置 eDP 时，只需正确配置背光、功率、GPIO 后，便可通过 EDID 点亮屏幕。由于屏幕的 EDID 有可能损坏，建议将 *edp\_timing* 配置到 dts 中，如上图所示。

## 备注

eDP 接口仅 SG368Z 系列模块支持。

### 3.3. HDMI

#### 3.3.1. HDMI 节点配置

在如下文件中配置 `status = "okay"`将`&hdmi`节点打开。

- Linux 操作系统:

Kernel 5.10 及之前: `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1: `kernel-6.1/arch/arm64/boot/dts/rockchip/display/rk3568-hdmi-out.dtsi`

- Ubuntu 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统:

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi`

```
&hdmi {
    status = "okay";
    rockchip,phy-table =
        <92812500 0x8009 0x0000 0x0270>,
        <165000000 0x800b 0x0000 0x026d>,
        <185625000 0x800b 0x0000 0x01ed>,
        <297000000 0x800b 0x0000 0x01ad>,
        <594000000 0x8029 0x0000 0x0088>,
        <000000000 0x0000 0x0000 0x0000>;
};
```

HDMI 的音频节点, 根据需要配置 `status = "okay"`或 `status = "disabled"`进行开关。

```
&hdmi_sound {
    status = "okay";
};
```

### 3.3.2. VOP 的挂载

选择要挂载的 VP，下图以 VP0 为例。当挂载到此 VP 下时，需要关闭其他的 VP 节点，因为一个 VP 下只能挂载一个显示接口，因此需要配置 `status = "disabled"` 将 `&hdmi_in_vp1` 节点关闭，然后配置 `status = "okay"` 将 `&hdmi_in_vp0` 节点打开，反之同理。

```
&hdmi_in_vp0 {                &hdmi_in_vp1 {
    status = "okay";           status = "disabled";
};                             };

```

由于 HDMI 需要独占 HPLL 的时钟源，此时钟源被挂在了 VP0 下，因此建议将 HDMI 挂载到 VP0 下，如果要将 HDMI 挂载到 VP1，还需要在如下文件中合入相关修改：

- Linux 操作系统：

Kernel 5.10 及之前： `kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi`

Kernel 6.1： `kernel-6.1/arch/arm64/boot/dts/rockchip/display/rk3568-hdmi-out.dtsi`

- Ubuntu 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi`

- OpenWrt 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi`

需合入的修改内容如下所示：

```
&vop{
    assigned-clocks = <&cru DCLK_VOP0>, <&cru DCLK_VOP1>, <&cru DCLK_VOP2>;
    assigned-clock-parents = <&cru PLL_VPLL>, <&pmucru PLL_HPLL>, <&cru PLL_GPLL>;
};

```

一般情况下，完成这些配置后，屏幕便可正常点亮。本章节仅介绍了 HDMI 的点亮流程，其他如 CEC、HDCP 等具体功能的配置可参考 SDK 的 `docs/Common/DISPLAY` 目录中的文档。

## 3.4. LVDS

### 3.4.1. LVDS 节点配置

在如下文件中配置 `status = "okay"` 将 `&lvds` 节点打开。

- Linux 操作系统:

Kernel 5.10 及之前: *kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi*

Kernel 6.1: *kernel-6.1/arch/arm64/boot/dts/rockchip/display/lvds0-single-channel-800x1280.dtsi*

- Ubuntu 操作系统:

*kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi*

- OpenWrt 操作系统:

*kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi*

```
&lvds{
    status = "okay";
};
```

### 3.4.2. VOP 的挂载

选择要挂载的 VP，下图以 VP2 为例，当挂载此 VP 下时，需要关闭其他的 VP 节点，因为一个 VP 下只能挂载一个显示接口，因此需要 *status = "disabled"* 将 *&lvds\_in\_vp1* 节点关闭，然后配置 *status = "okay"* 将 *&lvds\_in\_vp2* 节点打开，反之同理。

```
&lvds_in_vp2 {          &lvds_in_vp1 {
    status = "okay";      status = "disabled";
};                       };
```

### 3.4.3. lvds-panel 配置

在如下文件中找到 *lvds-panel* 节点进行 *panel* 属性配置。

- Linux 操作系统:

Kernel 5.10 及之前: *kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi*

Kernel 6.1: *kernel-6.1/arch/arm64/boot/dts/rockchip/display/lvds0-single-channel-800x1280.dtsi*

- Ubuntu 操作系统:

*kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi*

- OpenWrt 操作系统:

*kernel/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi*

```
lvds-panel{
    status = "okay";
    compatible = "simple-panel";
    backlight = <&backlight>;
    reset-gpios = <&gpio3 RK_PB6 GPIO_ACTIVE_HIGH>;
    enable-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
    reset-delay-ms = <120>;
    bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;

    width-mm = <217>;
    height-mm = <136>;

    display-timings {
        native-mode = <&lvds_timing>;
        lvds_timing: lvds_timing {
            clock-frequency = <75000000>;
            hactive = <800>;
            vactive = <1280>;
            hfront-porch = <46>;
            hsync-len = <2>;
            hback-porch = <44>;
            vfront-porch = <14>;
            vsync-len = <2>;
            vback-porch = <8>;
            hsync-active = <0>;
            vsync-active = <0>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
};
```

panel 的通用属性参考第 3.1.3 章的表 1 进行配置。LVDS 的 *bus-format* 与其他接口的不同，请参考数据映射和屏幕规格书的数据格式进行配置（详见第 3.4.4 章）。

### 3.4.4. 数据映射

*bus-format* 属性需要根据屏幕规格书中的数据格式进行配置，下图是屏幕规格书中的数据格式，需要在数据映射中找到对应的格式，填入至 *bus-format* 属性中。

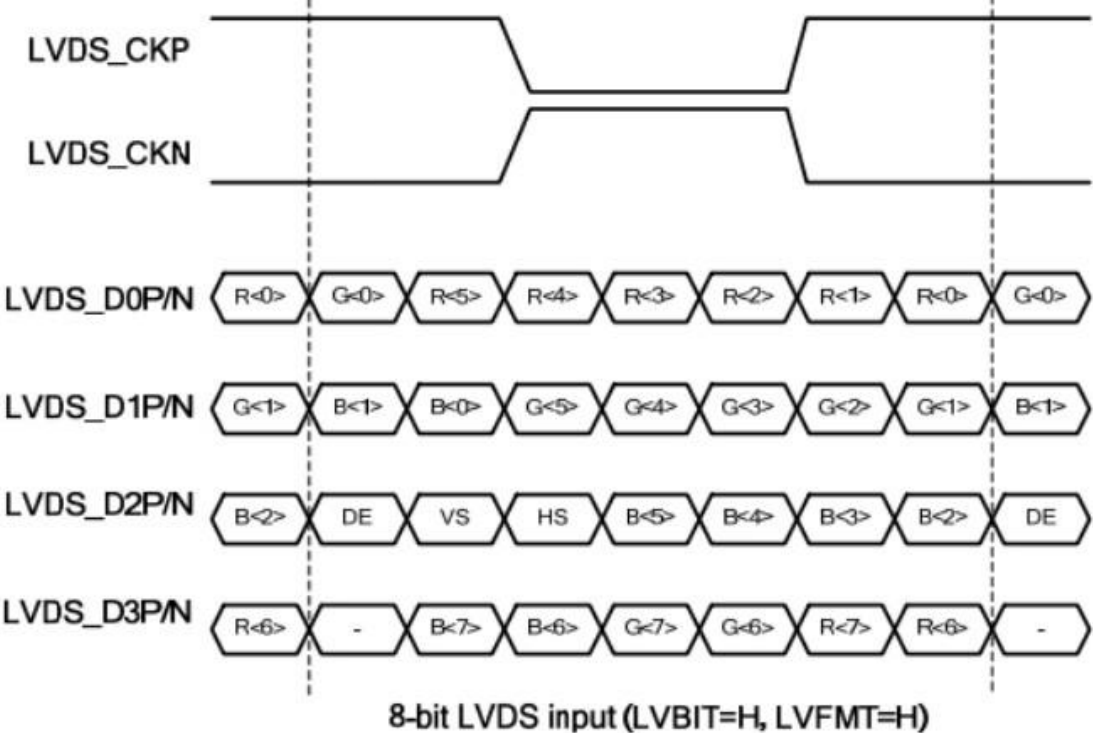


图 5：屏幕规格书数据格式

bus-format	Timeslot	Data organization			
		Lane3	Lane2	Lane1	Lane0
MEDIA_BUS_FMT_RGB666_1X7X3_SPWG	0	—	DEN	B1	G0
	1	—	VSYNC	B0	R5
	2	—	HSYNC	G5	R4
	3	—	B5	G4	R3
	4	—	B4	G3	R2
	5	—	B3	G2	R1
	6	—	B2	G1	R0
MEDIA_BUS_FMT_RGB888_1X7X4_SPWG	0	GND	DEN	B1	G0
	1	B7	VSYNC	B0	R5
	2	B6	HSYNC	G5	R4
	3	G7	B5	G4	R3
	4	G6	B4	G3	R2
	5	R7	B3	G2	R1
	6	R6	B2	G1	R0
MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA	0	GND	DEN	B3	G2
	1	B1	VSYNC	B2	R7
	2	B0	HSYNC	G7	R6
	3	G1	B7	G6	R5
	4	G0	B6	G5	R4
	5	R1	B5	G4	R3
	6	R0	B4	G3	R2

图 6：LVDS 数据映射

## 备注

关闭某个接口时，请将此接口上的所有节点都关闭，例如，关闭 MIPI 时，需要将 *dsi1*、*dsi1\_in\_vp0*、*dsi1\_in\_vp1*、*route\_dsi1* 和 *dsi1\_panel* 全部关闭。如果要更改某个接口的 VOP 时，对应的 VOP 不能与其他接口有关联。

---



# 4 显示功能配置

## 4.1. 背光配置

### 4.1.1. 背光节点属性介绍

如下以 eDP 为例介绍背光的配置流程，背光使用的节点是 *backlight2*（用于 Linux kernel 5.10 及之前和 Ubuntu 操作系统）或 *edp\_backlight*（用于 Linux kernel 6.1 操作系统）或 *backlight1*（用于 OpenWrt 操作系统），信息如下：

```
backlight2: backlight2 {
    compatible = "pwm-backlight";
    status = "okay";
    pwms = < &pwm7 0 25000 1 >;
    brightness-levels = <
        0 20 20 21 21 22 22 23
        23 24 24 25 25 26 26 27
        27 28 28 29 29 30 30 31
        31 32 32 33 33 34 34 35
        35 36 36 37 37 38 38 39
        40 41 42 43 44 45 46 47
        48 49 50 51 52 53 54 55
        56 57 58 59 60 61 62 63
        64 65 66 67 68 69 70 71
        72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87
        88 89 90 91 92 93 94 95
        96 97 98 99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
        168 169 170 171 172 173 174 175
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255
    >;
    default-brightness-level = <200>;
    enable-gpios = < &gpio1 RK_PD1 GPIO_ACTIVE_LOW >;
};
```

图 7：eDP 背光节点 1

```
edp_backlight: edp-backlight {
    compatible = "pwm-backlight";
    status = "okay";
    pwms = <&pwm7 0 25000 1>;
    brightness-levels = <
        0 20 20 21 21 22 22 23
        23 24 24 25 25 26 26 27
        27 28 28 29 29 30 30 31
        31 32 32 33 33 34 34 35
        35 36 36 37 37 38 38 39
        40 41 42 43 44 45 46 47
        48 49 50 51 52 53 54 55
        56 57 58 59 60 61 62 63
        64 65 66 67 68 69 70 71
        72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87
        88 89 90 91 92 93 94 95
        96 97 98 99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
        168 169 170 171 172 173 174 175
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255
    >;
    default-brightness-level = <200>;
    enable-gpios = <&gpio1 RK_PD1 GPIO_ACTIVE_LOW>;
    pinctrl-names = "default";
    pinctrl-0 = <&edp_bl_en_gpio>;
};
```

图 8: eDP 背光节点 2

表 3: 背光节点属性介绍

属性	描述
<i>pwms</i>	<ul style="list-style-type: none"> <li>● 参数 1, 表示引用的对应的 PWM 的节点</li> <li>● 参数 2, 表示 PWM 通道索引, 一般是 0, 因为 Rockchip PWM 每个芯片只有一个 PWM 通道</li> <li>● 参数 3, 表示 PWM 输出波形的时间周期, 单位: 纳秒。例如,</li> </ul>

	<p>上图配置的 25000, 表示期望的 PWM 输出周期是 2500 纳秒, 相应的频率为 1/2500 秒, 即 40 kHz</p> <ul style="list-style-type: none"><li>● 参数 4, 表示极性, 为可选参数</li></ul>
brightness-levels	<p>表示显示屏幕背光的亮度。通常以值 255 为一个标度, 表示亮度的范围从 0 到 255, 共 256 个元素</p> <ul style="list-style-type: none"><li>● 当 PWM 设置为正极性时, 从 0~255 表示背光为正极, 占空比从 0 % ~ 100 %变化</li><li>● 当 PWM 设置为负极性时, 从 255~0 表示背光为负极, 占空比从 100 % ~ 0 %变化</li></ul>
default-brightness-level	<p>表示默认元素的背光亮度。上图中配置为 200, 表示为第 200 个元素是屏幕的默认背光亮度</p>
enable-gpios	<p>表示背光使能引脚, 根据电路原理图配置即可; 若硬件没有此背光使能引脚, 则将此配置删除, 背光驱动通过配置 brightness-levels 数组的第 0 个元素将背光关闭</p>

4.1.2. 背光配置流程

根据如下原理图确认 eDP 背光使用的 PWM。结合原理图与 GPIO 配置表, 可以确认 eDP 的背光 PWM 使用的是 PWM7, 背光使能引脚是 GPIO1\_D1, 将上述信息填入 eDP 所引用的背光节点中即可完成背光配置。配置完成的背光节点见图 7 或图 8。

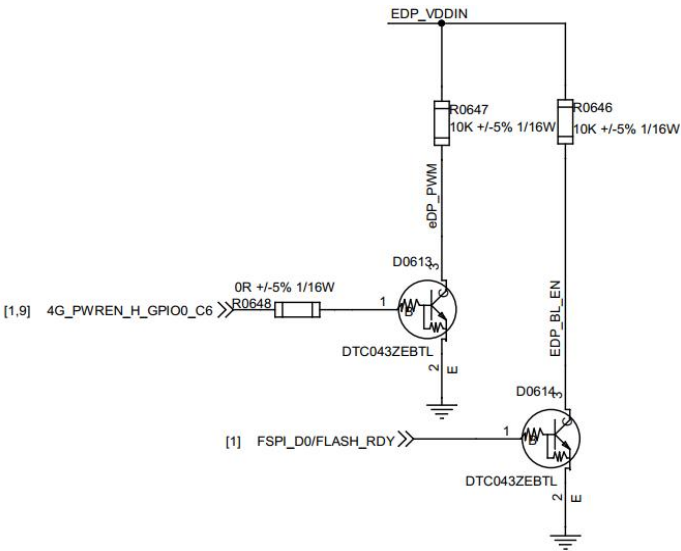


图 9: eDP 背光原理图

370	GPIO0_C6	d	GPIO0_C6_d	PWM7_IR	SPI0_CS0_M0	PCIEX2_PERSTn_M
267	GPIO1_D1	u	GPIO1_D1_u	FSPI_D0	FLASH_RDY	

图 10: GPIO 配置表

## 4.2. Logo 配置

### 4.2.1. Logo 节点介绍

Logo 通过如下文件中对应显示接口的 `route_xxx` 节点控制（U-Boot 显示组件和 Linux kernel 复用同一个 dtb），这里的 `xxx` 可以是 `dsi`、`edp`、`hdmi` 或 `lvds`，具体可搜索 dts 中的“route”关键字。

- Linux 操作系统：

Kernel 5.10 及之前： `kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

Kernel 6.1： `kernel-6.1/arch/arm64/boot/dts/rockchip/display/lvds0-single-channel-800x1280.dtsi`

- Ubuntu 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi`

- OpenWrt 操作系统：

`kernel/arch/arm64/boot/dts/rockchip/rk3568.dtsi`

下图分别是 `route_dsi1` 和 `route_lvds` 节点属性：

```
route_dsi1: route-dsi1 {
    status = "disabled";
    logo,uboot = "quectel_720p_fastboot.bmp";
    logo,kernel = "quectel_720p.bmp";
    logo,mode = "center";
    charge_logo,mode = "center";
    connect = <&vp0_out_dsi1>;
};
```

```
&route_lvds {
    status = "okay";
    connect = <&vp2_out_lvds>;
    logo,uboot = "logo.bmp";
    logo,kernel = "logo_kernel.bmp";
};
```

表 4: Logo 属性介绍

属性	描述
<i>logo,uboot</i>	描述 U-Boot 阶段 Logo 图片名称，需要和真实的图片资源匹配
<i>logo,kernel</i>	描述 Kernel 阶段 Logo 图片名称，需要和真实的图片资源匹配
<i>logo,mode</i>	控制 Logo 图片的显示模式，有 <i>center</i> （居中）和 <i>fullscreen</i> （全屏）两种模式
<i>charege_logo,mode</i>	充电时 Logo 图片的显示模式，有 <i>center</i> （居中）和 <i>fullscreen</i> （全屏）两种模式
<i>connect</i>	描述该显示接口和哪个 VOP 连接。一般根据当前显示接口挂载的 VP 进行配置

#### 4.2.2. Logo 节点配置

以 MIPI 为例，在 *kernel/arch/arm64/boot/dts/rockchip/display/rk3568-display.dtsi*（用于 Linux kernel 5.10 及之前和 Ubuntu 操作系统）或 *kernel/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10.dtsi*（用于 OpenWrt 操作系统）文件里找到 *route\_dsi1* 节点，将该节点的状态设置为 *status = "okay"*，*logo,uboot* 和 *logo,kernel* 一般根据当前要显示的 Logo 图片名称进行配置。*connect* 一般根据当前显示接口挂载的 VOP 进行配置，例如当前 *dsi1* 挂载在 VP1 下，则 *connect* 也同样要挂载在 VP1 下。

```
&route_dsi1 {
    status = "okay";
    logo,uboot = "quectel_720p_fastboot.bmp";
    logo,kernel = "quectel_720p.bmp";
    logo,mode = "center";
    connect = <&vp1_out_dsi1>;
};
```

以 LVDS 为例，在 *kernel-6.1/arch/arm64/boot/dts/rockchip/display/lvds0-single-channel-800x1280.dtsi*（用于 Linux kernel 6.1 操作系统）文件里找到 *route\_lvds* 节点，将该节点的状态设置为 *status = "okay"*，*logo,uboot* 和 *logo,kernel* 一般根据当前要显示的 Logo 图片名称进行配置。*connect* 一般根据当前显示接口挂载的 VOP 进行配置，例如当前 *lvds* 挂载在 VP2 下，则 *connect* 也同样要挂载在 VP2 下。

```
&route_lvds {
    status = "okay";
    connect = <&vp2_out_lvds>;
    logo,uboot = "logo.bmp";
    logo,kernel = "logo_kernel.bmp";
};
```

### 4.2.3. Logo 图片加载

Logo 节点配置完成后，需要将 *logo,uboot* 和 *logo,kernel* 中对应名称的 Logo 图片放入 *kernel/*目录下。同时还需要修改打包文件 *kernel/scripts/mkimf* 中 *LOGO\_KERNEL\_PATH* 和 *LOGO\_PATH*，将 Logo 编译到固件包内。

```
LOGO_PATH=${srctree}/quectel_720p_fastboot.bmp
[ -f ${LOGO_PATH} ] && LOGO=quectel_720p_fastboot.bmp

LOGO_KERNEL_PATH=${srctree}/quectel_720p.bmp
[ -f ${LOGO_KERNEL_PATH} ] && LOGO_KERNEL=quectel_720p.bmp
```

在 Linux kernel 6.1 中，Logo 节点配置完成后，需要将 *logo,uboot* 和 *logo,kernel* 中对应名称的 Logo 图片放入 *kernel-6.1/*目录下。同时还需要修改打包文件 *kernel-6.1/scripts/mkimf* 中 *LOGO\_KERNEL\_PATH* 和 *LOGO\_PATH*，将 Logo 编译到固件包内。

```
LOGO_PATH=${srctree}/logo.bmp
[ -f ${LOGO_PATH} ] && LOGO=logo.bmp

LOGO_KERNEL_PATH=${srctree}/logo_kernel.bmp
[ -f ${LOGO_KERNEL_PATH} ] && LOGO_KERNEL=logo_kernel.bmp
```

### 4.2.4. Logo 要求

1. U-Boot Logo 和 Linux Kernel Logo 分辨率相同，且分辨率必须是偶数；
2. 只支持 8-bit、16-bit、24-bit、32-bit 色彩深度的.bmp 格式图片；
3. U-Boot Logo 和 Linux Kernel Logo 必须同时开启，即 *logo.bmp* 和 *logo\_kernel.bmp* 必须同时提供，不能只提供其中一个。

## 5 附录 术语缩写

表 5：术语缩写

缩写	英文全称	中文全称
CEC	Consumer Electronics Control	消费电子控制
CPU	Central Processing Unit	中央处理器
DCS	Display Command Set	显示命令集
DDR	Double Data Rate	双倍数据率
DPI	Dots Per Inch	每英寸点数
DSI	Display Serial Interface	显示串行接口
DTS	Device Tree Source	设备树源文件
EDID	Extended display identification data	扩展显示标识数据
eDP	Embedded Display Port	嵌入式显示端口
EOTP	End of Transmission Packet	传输结束包
EVB	Evaluation Board	评估板
GPIO	General Purpose Input/Output	通用输入/输出
GPU	Graphics Processing Unit	图形处理单元
HDCP	High-bandwidth Digital Content Protection	高带宽数字内容保护
HDMI	High-Definition Multimedia Interface	高清晰度多媒体接口
HPD	Hot-Plug Detection	热插拔检测
HPLL	High-Speed Phase-Locked Loop	高速锁相环
ID	Identification	标识

LCD	Liquid Crystal Display	液晶显示屏
LP	Low Power	低功耗
LVDS	Low-Voltage Differential Signaling	低压差分信号传输
MIPI	Mobile Industry Processor Interface	移动产业处理器接口
PWM	Pulse Width Modulation	脉冲宽度调制
RGA	Rockchip Graphics Accelerator	瑞芯微图形加速器
SDK	Software Development Kit	软件开发工具包
VOP	Video Output Processor	视频输出处理器
VP	Video Port	视频端口
VPU	Video Processing Unit	视频处理单元